CS641A Mid Sem

Piyush Gangle, Punit Chaudhari, Aman Mittal

TOTAL POINTS

11 / 50

QUESTION 1

1 DES algorithm **0 / 25**

+ **5 pts** Discuss differentials at $$L_0R_0$$, $$L_1R_1$$, $$L_2R_2$$, and S-Boxes for the first 2 rounds.

+ **10 pts** Use the new S-Box design to conclude S-Box output XOR $$0000$$ occurs with probability $$\frac{32}{64}=\frac{1}{2}$$ for certain differentials

+ **5 pts** Mention the 2-round iterative characteristics with probability $$\frac{1}{2}$$

+ **3 pts** Use the above 2-round iterative characteristics to form 14-round characteristics with probability $$\frac{1}{128}$$

+ **2 pts** Discuss no. of pairs required to break this 16-round DES using $$p=\frac{1}{128}$$

✓ + **0 pts** **Wrong or NA**

QUESTION 2

2 Diffie Hellman **11 / 25**

+ **3 pts** State the existence of _disjoint cycles_ for a permutation $$p \in S_n$$

+ **5 pts** Describe a method to efficiently compute the _disjoint cycles_ of a permutation $$p$$

✓ + **2 pts** **State _disjoint cycles_ of the pair $$(g, g^c)$$ and/or $$(g, g^d)$$**

✓ + **10 pts** **Describe how to form a system of linear modular equations from the _disjoint cycles_ of $$(g, g^c)$$ and/or $$(g, g^d)$$ to compute $$c$$ and/or $$d$$**

✓ - **3 pts** **No explanation of why $$c \equiv r_i \, (mod \, l_i)$$ follows after finding the differences in positions**

✓ + **2 pts** **State how to compute $$c$$ or $$d$$ from the above equations**

+ **3 pts** Correctness of computed $$c$$ or $$d$$.

Use the fact that the order $$l$$ of $$g$$ is $$lcm$$ of the order of its _disjoint cycles_

+ **0 pts** Incorrect or NA

QUESTION 3

3 References **0 / 0**

✓ + **0 pts** **Correct**

CS641
Modern Cryptology
Indian Institute of Technology, Kanpur

Group Name: INSYNC
Punit Chaudhari (21111050), Piyush Gangle (21111046), Aman Mittal (180072)

Mid Semester Examination

Submission Deadline:
March 3, 2022, 23:55hrs

Question 1

Consider a variant of DES algorithm in which all the S-boxes are replaced. The new S-boxes are all identical and defined as follows.

> Let $b_1, b_2, \cdots, b_6$ represent the six input bits to an S-box. Its output is $b_1 \oplus (b_2 \cdot b_3 \cdot b_4), (b_3 \cdot b_4 \cdot b_5) \oplus b_6, b_1 \oplus (b_4 \cdot b_5 \cdot b_2), (b_5 \cdot b_2 \cdot b_3) \oplus b_6$.

Here '$\oplus$' is bitwise XOR operation, and '$\cdot$' is bitwise multiplication. Design an algorithm to break 16-round DES with new S-boxes as efficiently as possible.

Solution

DES stands for Data Encryption Standard it is a symmetric-key block cipher designed by the National Institute of Standards and Technology (NIST).DES uses 16 round Feistel structure. The block size is 64-bit. Though, key length is 64-bit, DES has an effective key length of 56 bits.
$\Longrightarrow$ Four major operations done in DES are :

- Expansion : Convert 32bit input into 48 bit output.
- XOR : output of Expansion $\oplus$ Round Key.
- S-box : 6bit input $\longrightarrow$ 4bit output.
- Permutation : Shuffle bits so that in all 4-bits in a block move to different blocks, for each of the eight blocks.

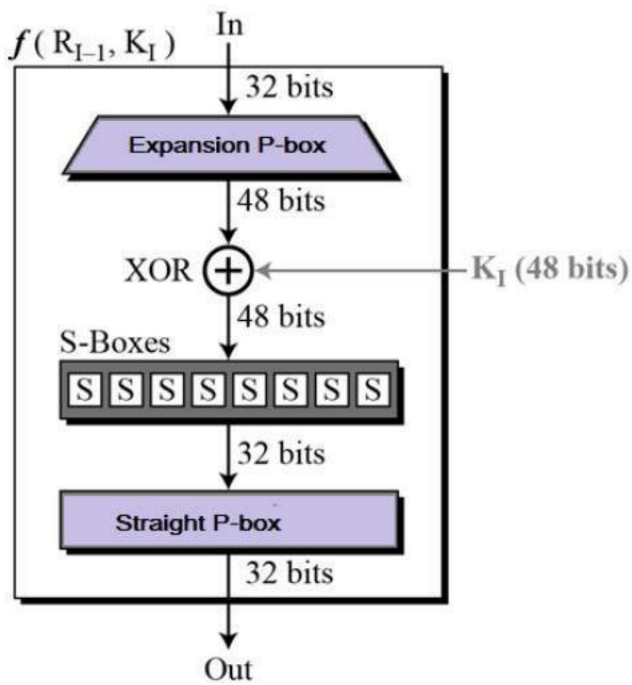

Figure 1: Diagrammatic illustration for DES round function

⟶ To break 16-round we are going to use iterative characteristics.let us consider 6 bit input to Sbox as $(b_1, b_2, b_3, b_4, b_5, b_6)$ and 4 bit output of S-box as (x_1, x_2, x_3, x_4).
⟶As Mentioned in given problem

1. $x_1 = b_1 \oplus (b_2 b_3 b_4)$
2. $x_2 = (b_3 b_4 b_5) \oplus b_6$
3. $x_3 = b_1 \oplus (b_4 b_5 b_2)$
4. $x_4 = (b_5 b_2 b_3) \oplus b_6$

⟶Result of following operation $x_1 \oplus x_3$ and $x_2 \oplus x_4$ we get

1. $x_1 \oplus x_3 = b_2 \cdot b_4 \cdot (b_3 \oplus b_5)$
2. $x_2 \oplus x_4 = b_3 \cdot b_5 \cdot (b_2 \oplus b_4)$

$\implies$ To further carry- out our analysis we consider some binary values for $x_1 \oplus x_3$ and $x_2 \oplus x_4$.

$\implies$Assumption 1: $x_1 \oplus x_3 = 1$

$x_1 \oplus x_3 = 1$ states that each of b_2, b_4 and $b_3 \oplus b_5$ are going to be 1. hence this further states that $x_2 \oplus x_4$ will be 0 (From $x_2 \oplus x_4 = b_3 \cdot b_5 \cdot (b_2 \oplus b_4)$). Therefore, we can uniquely determine the 6-bit strings $b_1b_2b_3b_4b_5b_6$ which generate 4-bit strings $x_1x_2x_3x_4$ fulfilling $x_1 \oplus x_3 = 1$ as follows:

1. $b_1 = x_1 \oplus b_3$
2. $b_2 = b_4 = 1$
3. $b_5 = 1 \oplus b_3$
4. $b_6 = x_2 \oplus 0 = x_4 \oplus 0$

Therefore given a 4-bit string $x_1x_2x_3x_4$ that satisfies $x_1 \oplus x_3 = 1$, we can have two such 6-bit strings that could have produced $x_1x_2x_3x_4$ (because we know that the 6-bit string can be uniquely determined by tuning b_3). The XOR of these two 6-bit strings can be given by: $(b_1b_2b_3b_4b_5b_6) \oplus (b_1'b_2'b_3'b_4'b_5'b_6') = 101010$

$\implies$Assumption 2: $x_2 \oplus x_4 = 1$

$x_2 \oplus x_4 = 1$ states that each of b_3, b_5 and $b_2 \oplus b_4$ are going be 1. Hence this further states that $x_3 \oplus x_5$ will be 0 (From $x_1 \oplus x_3 = b_2 \cdot b_4 \cdot (b_3 \oplus b_5)$). Therefore, we can find each b_i in the 6-bit strings $b_1b_2b_3b_4b_5b_6$ which produces 4-bit strings $x_1x_2x_3x_4$ satisfying $x_2 \oplus x_4 = 1$ as follows:

1. $b_1 = x_1 \oplus 0 = x_3 \oplus 0$
2. $b_3 = b_5 = 1$
3. $b_2 = 1 \oplus b_4$
4. $b_6 = x_2 \oplus b_4$

Therefore given a 4-bit string $x_1x_2x_3x_4$ which satisfies $x_2 \oplus x_4 = 1$, we can have two possible 6-bit strings that could have generated $x_1x_2x_3x_4$ (because we know that the 6-bit string can be uniquely determined by the choice of b_4). The XOR of these two 6-bit strings can be given by: $(b_1b_2b_3b_4b_5b_6) \oplus (b_1'b_2'b_3'b_4'b_5'b_6') = 010101$

$\Longrightarrow$Vulnerability Spot :

After brainstorming via above equations we can conclude that four bit strings for which both $x_2 \oplus x_4 = 1$ and $x_3 \oplus x_5 = 1$ can't be satisfied using the given S-box design. Over and above since there exist four such 4-bit strings (0011, 0110, 1001 and 1100), all possible 6-bit strings (i.e $2^6 = 64$) maps to 12 possible 4-bit strings.Further we can note that each 4-bit string for which either $x_2 \oplus x_4 = 1$ or $x_3 \oplus x_5 = 1$, can be uniquely produced from two possible 6-bit strings. Since there exist eight such 4-bit strings, there exist sixteen 6-bit strings that map to these eight 4-bit strings. Therefore the remaining 48 possible 6-bit strings map to 4-bit strings for which $x_2 \oplus x_4 = 0$ and $x_3 \oplus x_5 = 0$. Since there exist four such 4-bit strings (0000, 0101, 1010, 1111), the remaining 48 6-bit strings map to these 4-bit strings. We can take advantage of this loophole in-order to design an efficient 2-round iterative characteristic having high success rate.

$\Longrightarrow$Modeling Iterative Characteristic:

Now we move our focus to 2-round iterative characteristic of the form which is mentioned below:

$\longrightarrow$ $(00000000, 80000000, p, 80000000, 00000000, 1, 00000000, 80000000)$

Our next goal is to get value of p when $XOR(L_0) = 00000000$ and $XOR(R_0) = 80000000$. Representation which we are going to follow for output of first stage expansion block is defined as below: $u = u_1u_2u_3u_4u_5u_6u_7u_8$, the input of the S-box by $v = v_1v_2v_3v_4v_5v_6v_7v_8$ and the S-box output by $w = w_1w_2w_3w_4w_5w_6w_7w_8$ where $|u_i| = 6$, $|v_i| = 6$ and $|w_i| = 4$. With the choice of $L_0 \oplus L_0^{'}$ and $R_0 \oplus R_0^{'}$, we have:

1. $u_1 \oplus u_1^{'} = v_1 \oplus v_1^{'} = 010000$
2. $w_1 \oplus w_1^{'} = 0000$

$\longrightarrow$ Set X_i is defined as: $X_i = (v, v^{'}) | v \oplus v^{'} = v_1 \oplus v_1^{'}$ and $S(v) = S(v^{'})$

$\longrightarrow$The Key observation we noted was $|X_i| = 32$ that gives us $p = \frac{32}{64} \cong \frac{1}{2}$. Therefore iterating the characteristic in $(00000000, 80000000, p, 80000000, 00000000, 1, 00000000, 80000000)$ 7 times results a probability of $p_s = \frac{1}{128}$. Hence the number of plain-text pairs we must have in-order to break 16 round DES will be approx = $\frac{20}{p_s} \approx \mathcal{O}(10^3)$ which is far better and efficient than brute-force approach.

1 DES algorithm **0 / 25**

+ **5 pts** Discuss differentials at L_0R_0, L_1R_1, L_2R_2, and S-Boxes for the first 2 rounds.

+ **10 pts** Use the new S-Box design to conclude S-Box output XOR 0000 occurs with probability $\frac{32}{64}=\frac{1}{2}$ for certain differentials

+ **5 pts** Mention the 2-round iterative characteristics with probability $\frac{1}{2}$

+ **3 pts** Use the above 2-round iterative characteristics to form 14-round characteristics with probability $\frac{1}{128}$

+ **2 pts** Discuss no. of pairs required to break this 16-round DES using $p=\frac{1}{128}$

✓ + **0 pts** **Wrong or NA**

Question 2

Suppose Anubha and Braj decide to do key-exchange using Diffie-Hellman scheme except for the choice of group used. Instead of using F_p^* as in Diffie-Hellman, they use S_n, the group of permutations of numbers in the range $[1, n]$. It is well-known that $|S| = n!$ and therefore, even for $n = 100$, the group has very large size. The key-exchange happens as follows:

> An element $g \in S_n$ is chosen such that g has large order, say l. Anubha randomly chooses a random number $c \in [1, l-1]$, and sends g^c to Braj. Braj choses another random number $d \in [1, l-1]$ and sends g^d to Anubha. Anubha computes $k = (g^d)^c$ and Braj computes $k = (g^c)^d$.

Show that an attacker Ela can compute the key k efficiently.

Solution

The discrete logarithm problem is :

$\longrightarrow$Given a generator g and a group G, and if $h = g^a$ then we have to find a. The discrete logarithm is hard to solve when the group is specially chosen. For ex: Z_p^*, where the prime is specially chosen to be of the form $2 * r + 1$ where r is a prime. In this case, the group is chosen to be a symmetric group S_n. So, according to the properties of a symmetric group, let G_c be a cyclic group generated by g chosen from S_n. As an attacker, Ela knows g, g^a, g^b. Assuming that $h = g^a$, now to find **a = discrete log of h with base g**, we consider the following decompositions of h and g.

- $h = \beta_1 \circ \beta_2 \circ \beta_3 \circ \cdots \circ \beta_r$
- $g = \gamma_1 \circ \gamma_2 \circ \gamma_3 \circ \cdots \circ \gamma_s$

$\longrightarrow$ Here, all of β_i and γ_j are disjoint.
$\longrightarrow$ Now, we construct two arrays G and H using the following procedure:
For i = 1 to n:
    $G[i] = (j, pos(i))$ [means i is at pos(i) in the cycle γ_j]
For i = 1 to n:
    $H[i] = (k, pos(i))$ [means i is at pos(i) in β_k]

Note:

j $\longleftarrow$ index of i in cycle γ_j.

k $\longleftarrow$ index of i in cycle γ_k.

$\longrightarrow$This can be computed in O(n) time.

$\longrightarrow$Using the above decomposition, we effectively find $a \mod (|g|)$ where —g— is the order of G_c.

$\longrightarrow$Now we extract the first and second elements of each cycle of H and store them in the arrays A and B. Now use the following the following procedure to calculate the Difference array D:

For i = 1 to n: $D[i] = pos(B[i[) - pos(A[i])$ For cycles with a single element, we have the first and the second element same. So, $A[i] = B[i]$ This requires $O(n)$ time for searching elements as total cycle length is n.

Let array L = length of the cycle that contains i.

We now have the following r equations, which will lead to the calculation of a using Chinese remainder theorem.

$$
\begin{aligned}
a &\equiv x_1 \pmod{p_1} \\
a &\equiv x_2 \pmod{p_2} \\
&\vdots \\
a &\equiv x_r \pmod{p_r}
\end{aligned}
$$

Here, $p_1, p_2, \ldots, p_r$ need not be coprime(So, they can have common factors) Let the solution to these equations be a_r.

We can now calculate the solutions pairwise using Extended euclidean algorithm as follows:

$a = a_1 \mod (LCM(p_1, p_2))$

$a = a_3 \mod (p_3)$

We solve this in $O(\log(p1) * \log(p2)) = O(\log^2 n)$ time.

We finally solve $n - 1$ equations in

$$\mathcal{O}\left(\sum_{k=1}^{n-1} k \cdot \log^2 n\right) = \mathcal{O}\left(n^2 \log^2 n\right)$$

So, finally we get:
$a = a_r \mod (LCM(p_1, p_2, \ldots, p_r))$ in $\mathcal{O}\left(n^2 \log^2 n\right)$ time.

2 Diffie Hellman **11 / 25**

+ **3 pts** State the existence of _disjoint cycles_ for a permutation $$p \in S_n$$

+ **5 pts** Describe a method to efficiently compute the _disjoint cycles_ of a permutation $$p$$

✓ + **2 pts** **State _disjoint cycles_ of the pair $$(g, g^c)$$ and/or $$(g, g^d)$$**

✓ + **10 pts** **Describe how to form a system of linear modular equations from the _disjoint cycles_ of $$(g, g^c)$$ and/or $$(g, g^d)$$ to compute $$c$$ and/or $$d$$**

✓ - **3 pts** **No explanation of why $$c \equiv r_i \, (mod \, l_i)$$ follows after finding the differences in positions**

✓ + **2 pts** **State how to compute $$c$$ or $$d$$ from the above equations**

+ **3 pts** Correctness of computed $$c$$ or $$d$$. Use the fact that the order $$l$$ of $$g$$ is $$lcm$$ of the order of its _disjoint cycles_

+ **0 pts** Incorrect or NA

[Agr22b] [Agr22a] [Tut] [Dci]

References

[Agr22a] Dr. Manindra Agrawal. *Lecture 11 Modern Cryptology (CS641A)*. IIT Kanpur, 2022.

[Agr22b] Dr. Manindra Agrawal. *Lecture 4-7 Modern Cryptology (CS641A)*. IIT Kanpur, 2022.

[Dci] Towards Data Dcience. Diffie Hellman Key Exchange.

[Tut] Tutorialspoint. Data Encryption Standard.

3 References 0 / 0

✓ + 0 pts Correct